Encouraging reactivity to create robust machines

Joel Lehman, Sebastian Risi, David D'Ambrosio, and
Kenneth O. Stanley

To Appear In: *Adaptive Behavior* journal. London: SAGE, 2013.

Abstract

The robustness of animal behavior is unmatched by current machines, which often falter when exposed to unforeseen conditions. While animals are notably reactive to changes in their environment, machines often follow finely-tuned yet inflexible plans. Thus instead of the traditional approach of training such machines over many different unpredictable scenarios in detailed simulations (which is the most intuitive approach to *inducing* robustness), this work proposes to train machines to be *reactive* to their environment. The idea is that robustness may result not from detailed internal models or finely-tuned control policies but from cautious exploratory behavior. Supporting this hypothesis, robots trained to navigate mazes with a reactive disposition prove more robust than those trained over many trials yet not rewarded for reactive behavior in both simulated tests and when embodied in real robots. The conclusion is that robustness may neither require an accurate model nor finely calibrated behavior.

Keywords: evolutionary robotics, evolution of robustness, reactivity

1 Introduction

Among the distinctive hallmarks that separate natural organisms from machines is their robustness in the presence of uncertainty and unpredictability. Whether it is the lion slinking quietly over uncertain terrain or the cockroach fleeing the faintest vibration, animals exhibit a keen sensitivity and remarkable resilience to the most subtle variations. This fortitude naturally provides inspiration to researchers aiming to achieve similar robustness in the control of robotic machines. Yet interestingly, this goal remains elusive, especially in robot controllers trained through machine learning. Such controllers are notoriously brittle and unstable in the presence of noise (Brooks, 1994; Jakobi et al., 1995; Lipson and Pollack, 2000; Matarić, 1997; Matarić and Cliff, 1996; Miglino et al., 1995).

The primary obstacle to achieving robustness in learned robot controllers is that it is impossible to model precisely all details or every situation that can be encountered. As a result, robot controllers experience conditions outside the bounds of the models that governed their design when deployed in the real world, often causing undesirable behavior. In short, they usually learn inappropriately to depend upon idiosyncratic details encountered during training that may not be repeated once deployed in the real world (Jakobi et al., 1995).

The usual response to this problem is to attempt to *model* the real-world environment as closely as possible, which often includes modeling the distribution of noise in the environment that is likely to degrade the accuracy of sensory experience (Balakirsky et al., 2009; Michel, 2004; Ng et al., 2006; Nolfi and Floreano, 2000; Zufferey et al., 2006). This traditional approach reflects the philosophy that the *reason* organisms exhibit such remarkable robustness is that they are *highly tuned* to their environments through delicate neural control policies shaped over the eons of evolutionary selection on Earth (Baldwin, 1896; Hagen and Hammerstein, 2005).

In contrast, the aim of this article is to present and test an alternative hypothesis. The main idea is that rather than reflecting a delicately calibrated control policy, behavioral robustness (whether in nature or machines) may often result from policies that are optimized specifically to work in the presence of *inaccuracy* and poor internal modeling. In particular, a controller that is selected to be *reactive*, i.e. to continually seek out and react to changing information in the environment, naturally becomes robust because it explicitly *mistrusts* its model and thus searches relentlessly for clues to the real nature of its environment. The idea of training for such reactivity is motivated by the observation that biological organisms tend to probe their environments continually and react appropriately to changing stimuli (Berlyne, 1966; Glickman and Sroges, 1966) [1].

To test this hypothesis, a quantification of reactivity applicable to robots is derived from the *mutual information* statistic (Shannon, 1949), which has previously been applied in a different context to encouraging exploratory (though not in particular robust) behavior in robots (Ay et al., 2008). The new idea is that reactive robots exhibit a relationship between the intensity of environmental change and the intensity of their response, i.e. they noticeably exhibit that they pay attention to changes in their environment. Because such reactive robots seek and experience a greater variety of conditions, their success at a particular task may provide *more* evidence of their robustness than a similar success exhibited by a non-reactive robot. Supporting this hypothesis, experiments in this article with simulated and real wheeled robots demonstrate that robots trained to accomplish a navigation task while still behaving reactively prove more robust than robots trained more traditionally to solve the task through a variety of explicit models of environmental uncertainty (i.e. noise). Interestingly, robots trained to be reactive without any explicit model of environmental noise whatsoever are sometimes superior to those trained with such models.

The experiments presented here optimize populations of simulated robots (later transferred to the real world) through a biologically-inspired approach to robotics based on evolution. It is often the case in such evolutionary robotics

[1] This paper significantly expands on an initial preliminary conference paper on the idea of reactivity in Lehman et al. (2012). New content includes the first hybridization of reactivity and noise, more extensive experiments, and experiments with real robots.

(ER; Nolfi and Floreano, 2000) approaches that robustness is encouraged in an intuitive way by exposing robots to many instances of probabilistic simulations in which random noise is added to the robot's sensors and motors (Jakobi et al., 1995; Nolfi and Floreano, 2000). The intent is to devalue solutions dependent on merely circumstantial conditions. While this approach is logical and sometimes successful, robots in such noisy simulations, while not dependent on momentary idiosyncrasies, may learn to depend on *specific distributions* of noise. In contrast, reactivity means depending upon as few assumptions as possible.

In a final surprising result, *combining* reactivity with noisy simulation produces the most robust result while requiring multiple times fewer evaluations than traditional noise training. The overall conclusion is that robustness ultimately may neither require an accurate model nor finely calibrated behavior.

The next section provides background on the evolutionary approach to training. Section 3 then formalizes the notion of reactivity. The experimental design is detailed in section 4, followed by results in both simulation and the real world in section 5. The paper then concludes with final thoughts in section 6.

2 Background

This section reviews past work in evolving robust controllers in ER, the NEAT and HyperNEAT methods applied in the experiments, and multi-objective optimization.

2.1 Evolving for Robustness

For practical reasons, controllers for robots in ER are often trained in a computer simulation rather than directly in reality (Nolfi and Floreano, 2000). However, discrepancies between simulation and reality may cause controllers that are effective in simulation to fail when transferred to a real robot. Because this problem of crossing the reality gap is a significant issue in ER there exist specific training methods that attempt to mitigate it (Bongard and Lipson, 2004; Bongard et al., 2006; Jakobi, 1998; Koos et al., 2011; Zagal and Ruiz-Del-Solar, 2007). The reality gap is one facet of the larger difficulty of evolving general, robust controllers that are not overly dependent on simulation details (Pinville et al., 2011).

Nearly all training strategies for evolving robust controllers involve training at least some individuals with multiple trials (Gomez and Miikkulainen, 2004; Jakobi, 1998; Pinville et al., 2011), non-determinism (Gomez and Miikkulainen, 2004; Jakobi, 1998; Pinville et al., 2011), or evaluations in reality (Bongard and Lipson, 2004; Bongard et al., 2006; Koos et al., 2010, 2011; Zagal and Ruiz-Del-Solar, 2007; Zagal et al., 2004). A common motivation for noisy training is that real-world sensors often do experience some degree of noise; however, a deeper motivation is that strategically applying noise to a robot's sensors or effectors can prevent evolution from exploiting features specific to a particular simulation. In other words, evolution otherwise often learns to depend upon incidental features of the presented scenario that are not characteristic of the problem to be solved in general (Jakobi, 1998).

While the motivations may be reasonable, the computational cost of training with noise is significant because noisy evaluations normally consist of multiple trials to reduce uncertainty about a policy's average performance (Jakobi, 1998; Pinville et al., 2011). To reduce computational costs, some methods seek to evaluate only *some* individuals in a full suite of noisy trials by estimating transferability for other individuals (Pinville et al., 2011). Yet this approach still requires additional potentially expensive evaluations and the estimates of transferability may not always be accurate. In addition to computational costs, it is not always clear how many trials, in what distribution, and with what intensity noise should be applied in training to ensure successful transfer (Gomez and Miikkulainen, 2004). While Jakobi (1998) lays out a principled methodology based on *minimal simulations*, it still requires painstaking measuring and modeling to implement.

Other approaches leverage occasional evaluations of controllers in the real world to encourage or estimate transferability (Bongard and Lipson, 2004; Bongard et al., 2006; Koos et al., 2011; Zagal and Ruiz-Del-Solar, 2007). The main idea is that although time-consuming and potentially difficult to automate, such evaluations on physical robots can identify discrepancies between the simulator and reality. In this way, it is possible to co-evolve simulators and controllers to reduce discrepancies (Bongard and Lipson, 2004; Bongard et al., 2006; Zagal and Ruiz-Del-Solar, 2007), or to penalize behaviors that exploit them (Koos et al., 2011).

However, an interesting unexplored question is whether there exist distinguishing properties of robust robot controllers that are visible in a single simulated trial. If such properties exist and can be explicitly encouraged by an appropriate training incentive, it may be possible to evolve robust robot policies without multiple trials or intermittent evaluations in reality. While interesting in its own right, such a training methodology would also reduce computational cost and might reduce the need to model a domain precisely. To this end, the experiments in this paper explore incentivizing the *reactivity* of an evolved controller to encourage its robustness.

Thus these experiments require a method to evolve robot controllers. Though other methods could be applied, here the HyperNEAT neuroevolution method was chosen to optimize the robot controllers as a well-established representative method in ER (Clune et al., 2011; D'Ambrosio et al., 2012, 2011; Drchal et al., 2009; Gauci and Stanley, 2010; Haasdijk et al., 2010; Knoester et al., 2010; Stanley et al., 2009). The next section reviews the Neuroevolution of Augmenting Topologies (NEAT) approach, the foundation of HyperNEAT.

2.2 Neuroevolution of Augmenting Topologies

Because the idea in this paper is to optimize a measure of reactivity to encourage more robust behaviors, to perform the experiments an optimization method is needed. The NEAT method was originally developed to evolve artificial neural networks (ANNs) to solve difficult control and sequential decision tasks (Stanley and Miikkulainen, 2002, 2004). Evolved ANNs control agents that select actions based on their sensory inputs. Like the SAGA method (Harvey, 1993) introduced before it, NEAT begins evolution with a population of small, simple networks and *complexifies* the network topology into diverse species over generations, leading to increasingly sophisticated behavior. A similar process of gradually adding new genes has been shown in natural evolution (Martin, 1999).

However, a key feature that distinguishes NEAT from prior work in complexification is its unique approach to maintaining a healthy diversity of complexifying structures simultaneously, as this section reviews. Complete descriptions of the NEAT method, including experiments confirming the contributions of its components, are available in Stanley and Miikkulainen (2002), and Stanley and Miikkulainen (2004). This section briefly reviews the key ideas on which the basic NEAT method is based.

To keep track of which gene is which while new genes are added, a historical marking is uniquely assigned to each new structural component. During crossover, genes with the same historical markings are aligned, producing meaningful offspring efficiently. In traditional implementations of NEAT, speciation protects new structural innovations by reducing competition between differing structures and network complexities, thereby giving newer, more complex structures room to adjust. Networks are assigned to species based on the extent to which they share historical markings. It is important to note that this aspect of NEAT was altered in this paper to replace speciation in NEAT with an explicit genetic diversity objective, which achieves a similar effect. That way, NEAT is easily integrated into a multi-objective framework, as explained shortly. Finally, complexification, which resembles how genes are added over the course of natural evolution (Martin, 1999), is thus supported by both historical markings and protecting innovation, allowing NEAT to establish high-level features early in evolution and then later elaborate on them. In effect, then, NEAT searches for a compact, appropriate network topology by incrementally complexifying existing structure.

The next section reviews HyperNEAT, an extension of NEAT applied in the experiments as a representative example of a modern neuroevolution (i.e. evolving ANNs through evolutionary algorithms) method.

2.3 HyperNEAT

Many neuroevolution methods are *directly encoded*, which means each part in the phenotype is encoded by a single gene, making the discovery of repeating motifs expensive and improbable. Therefore, indirect encodings (Bongard and Pfeifer, 2003; Hornby and Pollack, 2002; Stanley and Miikkulainen, 2003) have become a growing area of interest in evolutionary computation and artificial life.

One such indirect encoding designed explicitly for neural networks is the Hypercube-based NeuroEvolution of Augmenting Topologies (HyperNEAT) approach (Gauci and Stanley, 2010; Stanley et al., 2009), which is an indirect extension of the directly-encoded NEAT approach (Stanley and Miikkulainen, 2002, 2004) reviewed in the last section. HyperNEAT has proven effective in a number of recent domains, including many-joint robot arm control (Woolley and Stanley, 2010), real-world Khepera robot control (D'Ambrosio et al., 2012, 2011), quadruped locomotion (Clune et al., 2011), checkers board evaluation (Gauci and Stanley, 2010), and robocup soccer (Verbancsics and Stanley, 2010). This section briefly reviews HyperNEAT; a complete introduction is in Stanley et al. (2009) and Gauci and Stanley (2010).

Rather than expressing connection weights as distinct and independent parameters in the genome, HyperNEAT allows them to vary across the phenotype in a regular pattern through an encoding called a *compositional pattern producing network* (CPPN; Stanley, 2007), which is like an ANN but with specially-chosen activation functions. Such CPPNs are used in HyperNEAT to represent the connectivity patterns of ANNs as a *function of geometry*. That is, if an ANN's nodes are embedded in a geometry, i.e. assigned coordinates within a space, then it is possible to represent its connectivity as a single evolved function of such coordinates. In effect the CPPN paints a pattern of weights across the geometry of a neural network. To understand why this approach is promising, consider that a natural organism's brain is physically embedded within a geometric space, and that such embedding heavily constrains and influences the brain's connectivity. Topographic maps (i.e. ordered projections of sensory or effector systems such as the retina or musculature) exist within brains that preserve geometric relationships between high-dimensional sensor and effector fields (Hubel and Wiesel, 1962; Udin and Fawcett, 1988). In other words, there is important information *implicit* in geometry that can only be exploited by an encoding informed by geometry.

In particular, geometric *regularities* such as symmetry or repetition are pervasive throughout the connectivity of natural brains. To similarly achieve such regularities, CPPNs exploit activation functions that induce regularities in HyperNEAT networks. The general idea is that a CPPN takes as input the geometric coordinates of two nodes embedded in the *substrate*, i.e. an ANN situated in a particular geometry, and outputs the weight of the connection between those two nodes (figure 1). In this way, a Gaussian activation function by virtue of its symmetry can induce symmetric connectivity and a sine function can induce networks with repeated elements. Note that because the size of the CPPN is decoupled from the size of the substrate, HyperNEAT can compactly encode the connectivity of a an arbitrarily large substrate with a single CPPN. In short, in HyperNEAT, NEAT evolves CPPNs that compactly encode larger ANNs.

It is important to note that HyperNEAT is chosen here simply as a representative modern neuroevolution method. Because all experiments are based on HyperNEAT, the main distinctions among them will be the use of noise or reactivity in training rather than the training algorithm or its particular details.

The next section reviews multi-objective optimization, which is combined with HyperNEAT to enable optimizing both reactivity and fitness during a single run.

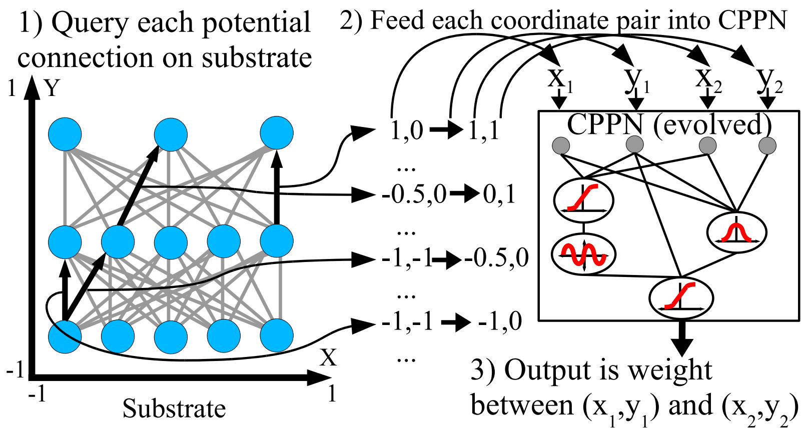

1) Query each potential connection on substrate
2) Feed each coordinate pair into CPPN
1
Y
-1
-1
X
1
Substrate
1,0 → 1,1
...
-0.5,0 → 0,1
...
-1,-1 → -0.5,0
...
-1,-1 → -1,0
...
x1
y1
x2
y2
CPPN (evolved)
3) Output is weight between (x1,y1) and (x2,y2)

Figure 1: **CPPN-based Geometric Connectivity Pattern Encoding.** A collection of nodes, called the *substrate*, is assigned coordinates that range from −1 to 1 in all dimensions. (1) Every potential connection in the substrate is queried to determine its presence and weight; the dark directed lines in the substrate depicted in the figure represent a sample of connections that are queried. (2) Internally, the CPPN (which is evolved by NEAT) is a graph that determines which activation functions are connected. As in an ANN, the connections are weighted such that the output of a function is multiplied by the weight of its outgoing connection. For each query, the CPPN takes as input the positions of the two endpoints and (3) outputs the weight of the connection between them. Thus, CPPNs can produce regular patterns of connection weights in space.

2.4 Multi-objective Optimization

Multi-objective optimization is a popular paradigm within EC that addresses how to optimize more than one objective at the same time in a principled way (Coello, 1999). The experiments in this paper apply an implementation of NSGA-II (Deb et al., 2002), a well-established Pareto-based multi-objective search algorithm, to optimize a traditional fitness objective and a reactivity objective concurrently.

The concept of dominance is central to Pareto-based multi-objective search; the key insight is that when comparing the performance of two individuals over multiple objectives, if both individuals are better on different subsets of the objectives then there is no meaningful way to directly rank such individuals because neither entirely *dominates* the other. That is, ranking such mutually non-dominating individuals would require placing priority or weight on one objective at the cost of another; traditionally one individual dominates another only if it is no worse than the other over all objectives and better than the other individual on at least one objective.

In this way, the best individuals in a population are those that are not dominated by any others. Such best individuals form the *non-dominated front*, which defines a series of trade-offs in the objective space. That is, the non-dominated front contains individuals that specialize in various combinations of optimizing the set of all objectives. Some will maximize one at the expense of all the rest, while some may focus equally on all of the objectives. The result is that various trade-offs of competing objectives such as genomic diversity, fitness, and reactivity can be explored during a single evolutionary run. The hope is that particular trade-offs between fitness performance and reactivity (i.e. policies that perform as well as possible given the constraint that they must be reactive) may lead to more robust behavior. Interestingly, such trade-offs may also mitigate the potential for a reactivity objective to discourage more cognitive controllers that are not *always* reactive; that is, if temporarily ignoring sensor inputs is necessary to increase performance, then the concept of non-dominance implies that they can yet survive in the population.

Note that adding a reactivity objective can be seen as an example of *multi-objectivation* (Coello, 2006; Jensen, 2003; Knowles et al., 2001), where an additional objective unrelated to directly solving the problem can nonetheless aid search. The ability of such helper objectives in ER to encourage robustness and consistency of behaviors has been previously demonstrated (Koos et al., 2011; Ollion et al., 2012; Pinville et al., 2011), supporting the motivation of the multi-objective approach here.

Finally, recall that a detail of combining NEAT or HyperNEAT with multi-objective optimization is that NEAT has a mechanism (called speciation) for preserving genomic diversity that does not fit naturally into NSGA-II. Thus in the experiments in this paper, speciation is replaced in NEAT with an explicit *genomic diversity objective* that is similar in spirit. In particular, the genomic diversity of a given genome is quantified as the average distance to its k-nearest neighbors in genotype space as measured by NEAT's genomic distance measure. In this way, multi-objective evolution with NEAT is incentivized to maintain genomic diversity in a similar way to how it is in the original formulation of NEAT.

The next section formalizes the measure of reactivity that will be used as an additional objective for training.

3 Training for Reactivity

The hypothesis in this paper is that an agent that is more reactive to its environment will also be more robust. This view is in part inspired by the fluid reactivity and curiosity of natural organisms (Berlyne, 1966; Glickman and Sroges, 1966), which may relate to their robustness. Thus a promising idea is to encourage reactivity in ANN controllers for robots for two reasons: 1) to probe whether reactivity may indeed contribute to biological robustness by isolating it and applying it to an artificial context, and 2) to explore the practical issue of whether encouraging reactivity can increase the robustness of machines. For example, a robot exploring a maze that is continually probing and reacting to the walls with its rangefinder sensors may be more robust than a robot that always executes an inflexible memorized plan (which could be disrupted easily by unexpected noise in its sensors or effectors). However, to directly optimize reactivity so that it can be encouraged to evolve, it needs to be quantified.

In this article the notion of reactivity is formulated as a measure of statistical dependence between the magnitude of changes in a robot's sensors and its effectors. In general, dependence between two variables implies a consistent relationship between them (e.g. an increase in one variable may tend to result in a decrease in the other). More specifically, it implies that knowledge of one variable *helps* to predict the other. Encouraging such dependence makes sense because it provides evidence that an agent is paying *attention* to changes in its immediate situation. In particular, it implies that the magnitude of change in a robot's sensors influences the magnitude of change of its effectors. In this way, the measure is agnostic to the exact relationship between the two because the ideal such relationship may vary between domains. However, it ensures at least that reactions to sensory changes are consistent, which aligns well with the idea of reactivity.

For example, a particularly attentive student might nod vigorously when an important concept is explained but only slightly when a trivial theorem is proven. In contrast, for a blind person navigating with a cane in a corridor, a sudden large change in distance from the wall may call for caution and only gradual adjustment. Although such a consistent nodding or adjustment policy might not be directly *necessary* to solve the task, it provides *evidence* that the behavior is reactive. This evidence is the key to the success of the hypothesis in this paper: Even if forcing agents to provide evidence that they are paying attention slightly slows down behaviors that might otherwise be faster, it is still worth that cost for the robustness it buys in the end. For this purpose, the proposed measure of statistical dependence is that of *mutual information* (Shannon, 1949). The choice of this measure is also justified by past experiments in which mutual information incentivized simple exploratory behavior in robots (Ay et al., 2008), though that work was not focused on encouraging robustness.

The mutual information statistic for two continuous random variables takes the form

$$I(X;Y) = \int_Y \int_X p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dx\ dy, \tag{1}$$

where $p(x,y)$ is the joint probability distribution function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y. The higher the absolute value of $I(X;Y)$, the more dependent are the two variables. In particular, mutual information is maximized when the entropy (i.e. uncertainty) of considering X and Y independently is maximized but the entropy of considering X conditional on prior knowledge of Y (or vice versa) is minimized.

For the experiments that follow, reactivity is measured by the mutual information between the magnitude of changes in a robot's rangefinder sensors and the magnitude of changes in its motor effectors. However, this approach is general enough to be applied to different sensory setups in robots in other ER domains where probing and reacting is also important to robustness. Formally, the seven rangefinder sensors $i_1, .., i_7$ of the simulated robot are subtracted from their values on the previous timestep and the average magnitude of these *differences* at timestep t is recorded as x_t. The average change in the robot's outputs y_t is computed accordingly.

Because the true distributions of X and Y are not known, $p(x)$, $p(y)$, and $p(x, y)$ are estimated through histograms (with a bin width of 0.05) of the sampled data x_t and y_t collected during an evaluation. That is, three histograms are created: two one-dimensional histograms (one over x_t for $p(x)$ and one over y_t for $p(y)$), and one two-dimensional histogram (over both x_t and y_t for $p(x, y)$). Riemann sums are then applied to approximate the integrals from equation 1. However, any reasonable means of estimating the distributions or of numerical integration could be substituted.

An important insight in the proposed approach is that while optimizing this formalized measure of reactivity *alone* would not necessarily lead to successful task performance, instead it can augment training performance as an *additional objective* by employing a multi-objective optimization algorithm (Deb et al., 2002). In this way, individuals might be evolved that both solve a given task and provide evidence of potential robustness by being reactive, without the need for multiple noisy trials. The motivation is that if robust solutions could be evolved through this approach, computational costs would be reduced, as would the need for precisely modeling the domain (including estimating appropriate distributions of noise).

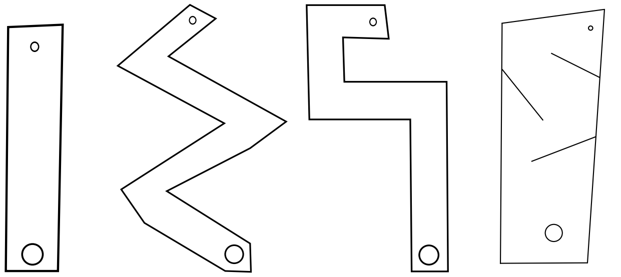
The next section describes experiments designed to explore the effect of reactivity on robustness and contrast it with more traditional approaches.

4 Maze Navigation Experiments

Because reactivity is intended to encourage robust behaviors, a domain for testing reactivity should be challenging under noisy conditions. For this purpose four maze navigation domains (figure 2) that create such a challenge in different ways are explored in this article.

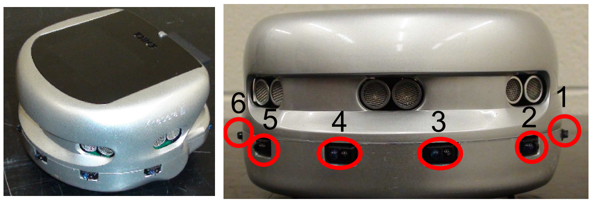
4.1 Maze Navigation Domains

In all of the mazes, the simulated robot is modeled after the Khepera III (K-Team, 2010) (shown in figure 3), which is the model used in the real-world portion of the experiments, and training and testing noise levels are in line with established models of the robot (Cyberbotics, 2012). An evolved ANN controls the robot with the goal of navigating from a starting point to an end point in a fixed time limit that requires direct traversal. To sense its environment, the robot has six rangefinders that indicate the distance to the nearest obstacle. Its three effectors produce forces that respectively turn and propel the robot.



(a) Straight (b) Zigzag (c) Winding (d) Deceptive

Figure 2: **Domains.** The goal of the agent in the maze navigation domains is to navigate from the starting position (large circle) to the goal (small circle). Note that mazes are not drawn to scale.



(a) Khepera III (b) Front Sensors

Figure 3: **Khepera III with Korebot II.** The Khepera III mobile robots (a) in these experiments (which are trained in simulation and later transferred to the real world) come equipped with a Korebot II extension that runs an embedded Linux operating system and allows the robots to receive broadcast communications over a wireless network. Although the Khepera III has many sensors available, only the front six infrared rangefinders (b) are utilized in these experiments.

In the Noise training variations, noise was applied to both the simulated robot's rangefinder sensors and its motor outputs. Such noise was computed according to the weighted average $(1.0 - x)v + xn$, where x is the noise level, v is the before-noise value, and n is randomly chosen from the unit uniform distribution.

The first domain, the Straight maze (figure 2a), is designed to be simple but yields situations that only become *necessary* to experience when an evolved behavior is exposed to significant levels of noise. That is, although an unconditional "always go forwards" policy will be effective without noise, sufficient effector noise may cause the robot's heading to veer into walls. To further accentuate such situations, in this maze the robot is disabled for the remainder of a trial if it collides with a wall. The Zigzag maze (figure 2b) is slightly more complicated because of the need to turn, but it and the remaining mazes allow the robot to recover if it hits a wall. The Winding maze (figure 2c), with its right-angle turns and narrower corridors, creates significant opportunity for the robot to get stuck or confused with increasing noise. Finally, the most challenging maze, the Deceptive maze (figure 2d), has deceptive cul-de-sacs that may complicate training in addition to sharp corners that are difficult to navigate with noise.

Four different ER approaches are compared to investigate the potential of training for reactivity:

- In the **Standard** setup there is a single *deterministic* trial (i.e. the robot performs ideally in the sense that there is no deviation from expected responses in its sensors and motors). ANN controllers for robots are optimized towards increasing success in accomplishing the domain task. Robots trained with this approach are expected to generalize poorly because no attempt is made to model noise in the environment.

- In the three **Noise** setups the optimization criteria remain the same as in the standard setup, but each robot is evaluated in *eight* non-deterministic noisy trials to provide a more realistic estimate of its performance in the real world. The amount of both sensor and effector noise for the three different noise setups is respectively 10%, 20%, and 30%, applied applied as follows: Noise is computed according to the weighted average $(1.0 - x)v + xn$, where x is the noise level, v is the before-noise value, and n is randomly chosen from the unit uniform distribution. Of course, the training noise level closest to the noise level in testing would be expected to yield the best performance.

- In the **Reactivity** setup an additional optimization criterion for increased reactivity complements the objective of accomplishing the domain task. As in the Standard setup, the robot is evaluated only in a *single* deterministic trial with no noise. The interesting question is whether a robot trained in such a setup, without any attempt to model noise, would nevertheless perform as well or better than those trained for a noisy environment.

- The **Reactivity + Noise** setup follows the Reactivity setup but each robot is evaluated in a *single* non-deterministic noisy trial to determine its quality. The amount of both sensor and effector noise for the three different Reactivity + Noise setups is respectively 10%, 20%, and 30%, following the Noise setup. This setup explores whether reactivity complements training with noise by significantly reducing the number of noise trials (from eight down to one) needed to learn effectively from noise.

4.2 Optimization Details

For these experiments HyperNEAT was combined with a multi-objective approach based on the popular NSGA-II algorithm (Deb et al., 2002). All ex-

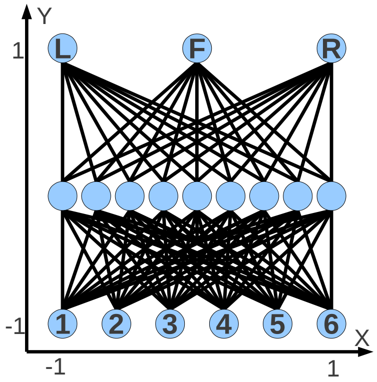


Figure 4: **HyperNEAT ANN Substrate.** The substrate ANN that HyperNEAT evolves is shown. The labeled numbers indicate the input nodes for the six rangefinder sensors, the unlabeled nodes are the hidden nodes, and the L, F, and R nodes are the output nodes for the left, forward, and right effectors of the robot, respectively.

periments optimize a primary objective that estimates progress towards accomplishing the domain task (i.e. navigating through a maze). Some experiments also concurrently optimize the reactivity of robots to investigate the hypothesis that reactivity may increase robustness. A final objective included in all experiments encourages population diversity (D'Ambrosio et al., 2012). The idea is to reward exploring a range of genotypes to avoid converging to an ANN controller that is only locally optimal and does not successfully solve the task. In HyperNEAT, sensors and effectors are placed in a geometric arrangement called the *substrate* to reflect the geometry of sensors and effectors on the robot in the real world. The neural substrate for the robots in this experiment is shown in figure 4, and is adapted from similar substrates used in past experiments with Khepera robots (D'Ambrosio et al., 2011).

The fitness of an individual is calculated as its distance to the goal at the end of the evaluation, which is a standard measure of progress in maze navigation tasks (Iba and Terao, 2000; Langdon et al., 1999; Lehman and Stanley, 2011; Mouret and Doncieux, 2012; Revello and McCartney, 2000). Runs of the Straight, Zigzag, and Winding mazes lasted 400 generations, while because of its increased difficulty runs of the Deceptive maze lasted $1,000$ generations.

The experiments were run with a modified version of the public domain SharpNEAT package (Green, 2006). The size of each population was 250 with 20% elitism. Asexual offspring (50%) had 0.96 probability of link weight mutation, 0.03 chance of link addition, and 0.01 chance of node addition. The coefficients for determining genomic similarity were 1.0 for nodes and connections and 0.1 for weights. The available activation functions were sigmoid, Gaussian, absolute value, and sine. Parameter settings are based on standard SharpNEAT

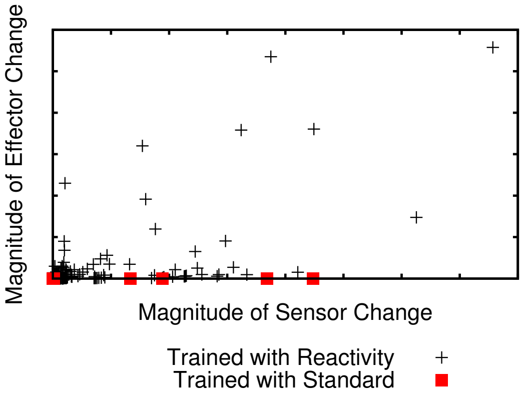


Figure 5: **Reactivity Calculation for Representative Controllers.** A scatter plot of paired magnitudes of sensor and effector changes for each time step of the simulation is shown for representative controllers from the Reactivity and Standard setups. The ANN controller trained with Reactivity receives a higher reactivity score because there is a higher mutual information between the two plotted dimensions. In particular, there is greater uncertainty in the independent distributions of sensor and effector change magnitudes, and less uncertainty in the conditional distributions.

defaults and were found to be robust to moderate variation through preliminary experimentation.

4.3 Example Trajectories and Reactivity Measures

To illustrate how reactivity is detected, figure 5 shows a scatter plot of the relevant dimensions for calculating mutual information of two characteristic behaviors in the Straight maze. Note that in the Reactivity example, there is more variance in both plotted dimensions, and knowing one dimension generally helps predict the other. This increase in prediction accuracy of one dimension from knowing the other, which is higher for the Reactivity example, is the mutual information between the two dimensions. Figure 6 shows the trajectories of these same individuals, demonstrating the link between reactive behavior and the reactivity measure.

5 Results

This section first presents training results and then turns to generalization, real-world transfer, and transfer to novel environments.

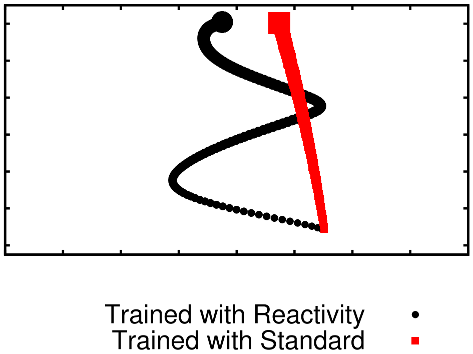


Figure 6: **Trajectory in Straight Maze of Representative Controllers.** The trajectories of the controllers from figure 5 are shown when they are evaluated in the Straight maze simulation. The ANN controller trained with Reactivity exhibits a more exploratory behavior that generates a greater variety of sensor and effector changes.

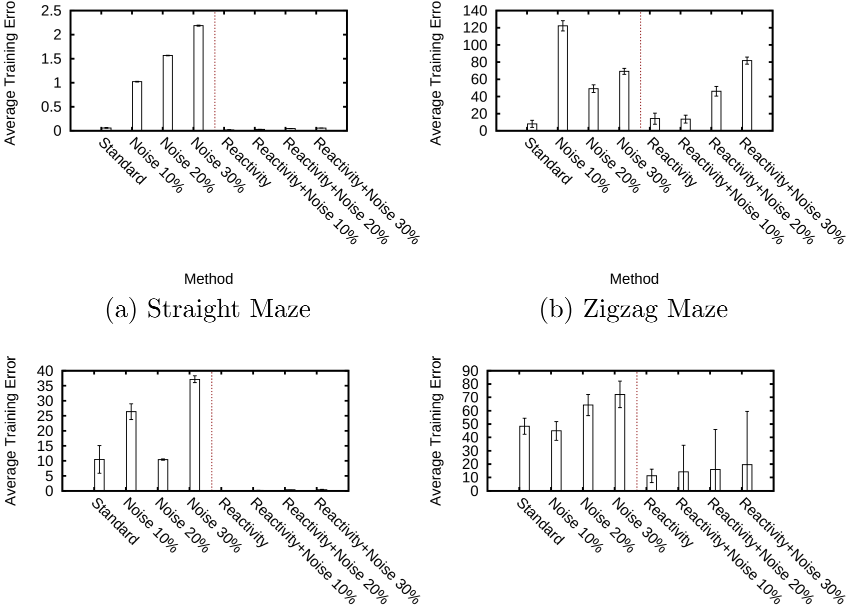
5.1 Training Results

The results of the *training* performance are shown in figure 7. The Reactivity setups with and without noise reach a significantly lower training error than any of the Noise setups or the Standard setup in the Deceptive and Straight mazes ($p < 0.05$; Student's t-test). Reactivity also achieves lower training error in the Winding maze in all but the 10% Noise setup ($p < 0.01$; Student's t-test). Additionally, the Standard setup actually outperforms some of the Noise setups in each of the four mazes ($p < 0.05$; Student's t-test), highlighting the potential for the noise model itself to increase the difficulty of training.

These results thus support the hypothesis that training with noise *alone* may often complicate training. However, *training* performance may not reflect robustness to noise; the Standard and Reactivity setups in fact both had no exposure to noise at all. It is important to note that even when a complete solution is not evolved in training, a partial evolved solution might still sometimes solve the task in the more lenient generalization test that is described next.

5.2 Generalization Test

Because the motivation for this experiment is to investigate the robustness of evolved controllers, a generalization test was devised to measure how well an evolved controller would perform in noisy distributions not encountered during training. The generalization test consists of 50 noisy trials with the length of evaluation doubled from training to allow for greater lenience. Such lenience reflects that in transfer slight stumbles due to the difference between the real world and the training environment are preferred to catastrophic failures (i.e. when a policy can never solve the task irrespective of how much time is allot-



(a) Straight Maze

(b) Zigzag Maze

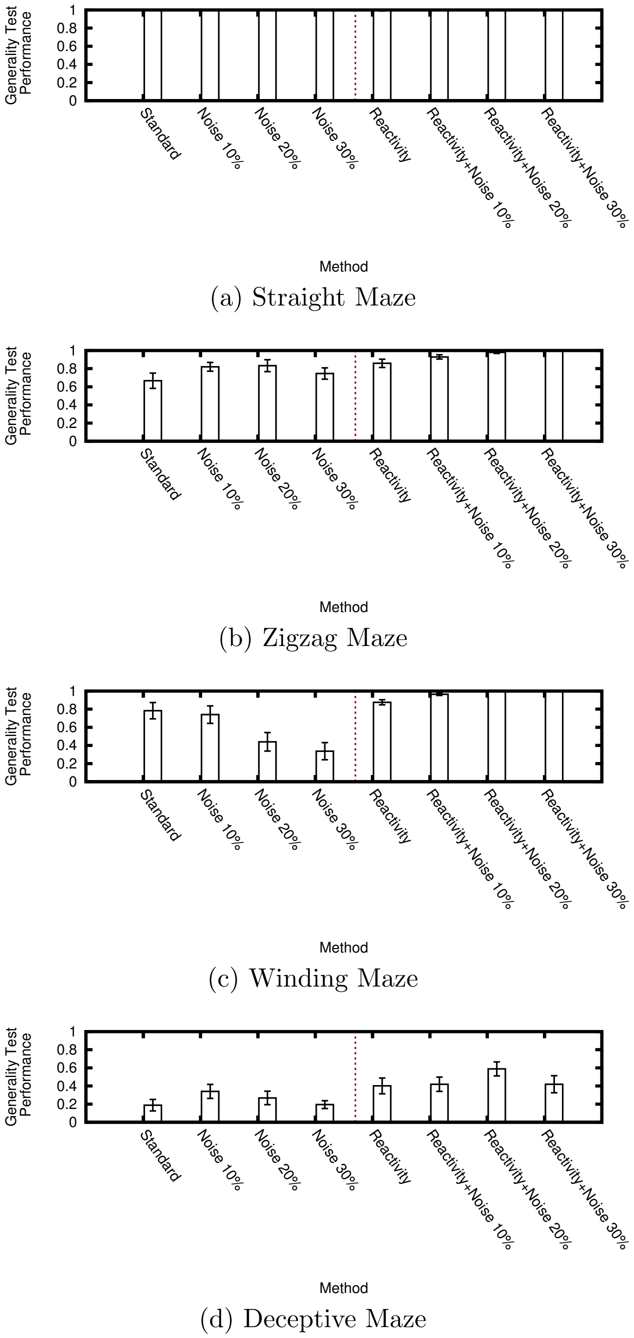
(c) Winding Maze

(d) Deceptive Maze

Figure 7: **Maze Navigation Training Error Results.** The average training error, i.e. the closest distance to the goal achieved by the best-performing individual in each run, is shown for the different training methodologies averaged over twenty independent runs. Generally the Reactivity setups perform competitively even though optimization in such setups must balance two different objectives.

ted). For each of the 50 trials, the sensor and effector noise levels are sampled independently from a uniform distribution ranging from 5% to 35%. The idea is to estimate the robustness of an evolved robot controller over a wide yet reasonable range of noisy situations.

An individual receives a score on the generalization test in accordance with the fraction of trials in which it is able to navigate the maze successfully (if it comes closer than 20 centimeters to the goal at any point). For each run, the individual scoring the overall highest on this test in the final population is recorded, and the average over each of the 20 runs reflects a setup's performance in the test. This approach to testing gives a sense of the most robust ANN controller one can hope to find with each approach. Figure 8 shows the results of the generalization test.



(a) Straight Maze

(b) Zigzag Maze

(c) Winding Maze

(d) Deceptive Maze

Figure 8: **Maze Navigation Generality Test Results.** The average probability of the best individual from a run solving the generality test is shown for different training methodologies over the four maze domains.

5.2.1 Generalization Performance of Reactivity Alone and Noise Setups

This section examines how the non-reactive setups (Standard and the three Noise setups) compare to training with the Reactive setup on the generalization test. Over these five compared setups with four different mazes, there are 20 total generalization scenarios and 16 pair-wise comparisons with the Reactivity setup. Overall, the Reactivity setup was never significantly worse than the more traditional setups and was often significantly better (in 5 out of 16 total pair-wise comparisons; Student's t-test; $p < 0.05$).

Due to its simplicity, the Straight maze did not distinguish any of the setups in this generalization test (although it does in the fine-grained generalization test described later). However, the other mazes proved more informative in separating the setups. In particular, it is interesting to explore whether Reactivity provides any advantage over the Standard setup. Supporting its motivation, the Reactivity setup outperforms the Standard setup in both the Zigzag and Deceptive mazes and is never outperformed by Standard. Furthermore, despite having no exposure to noise during training, the Reactivity setup never performs significantly worse than the Noise setups. In fact, Reactivity alone outperforms 20% and 30% Noise in the Winding maze, and also outperforms the 30% Noise setup in the Deceptive maze. These results suggest that training with some noise models can hurt robustness while reactivity can sometimes circumvent the need for choosing a particular noise model at all.

5.2.2 Generalization Performance of Reactivity + Noise

Results so far suggest that training with reactivity often performs as well as training explicitly with noise, and is sometimes significantly better. An interesting question is whether the performance of reactivity can be further improved if the robot trained for reactivity is evaluated in a single non-deterministic *noisy* trial. These results are also shown in Figure 8. Overall, the Reactivity + Noise setups most often perform significantly better than the other approaches (in 36 out of 60 pairwise comparisons) and are never significantly worse.

The Straight maze, as in the previous section, provides no useful distinctions for this generalization test. However, in both the Zigzag and Winding mazes, the Reactivity + Noise setups nearly always outperform the other approaches (in 28 out of 30 pair-wise comparisons). Finally, in the Deceptive maze, all of the Reactivity + Noise setups outperform the Standard setup although they do not outperform Reactivity alone. Reactivity + Noise in this maze also most often outperforms the Noise setups (in five out of nine pair-wise comparisons). These results demonstrate that evaluating robots in *one* noisy trial combined with reactivity is a promising new approach for training for robustness that relinquishes the need for multiple noisy trials.

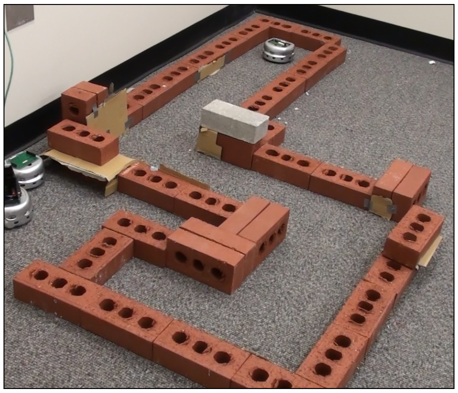


Figure 9: **Winding Maze.**

5.3 Transfer to the Real World

To validate the potential benefits of reactive behaviors for crossing the reality gap (Jakobi et al., 1995) to the real world, this section presents results of real-world transfers. The Winding maze was chosen as a test environment for transfer because it was one of the more challenging maze setups in training (figure 7) and was also designed to be easily realizable in the real world. The maze was constructed out of red $7\frac{5}{8}$in $\times 3\frac{5}{8}$in$\times 2\frac{1}{4}$in bricks with a carpet base (figure 9), which are the same dimensions as in the simulator.

The generalization test described in the previous section decided which individuals to transfer for each method. In particular, the best-scoring individual on the Winding maze generalization test from each of the 20 independent runs for each method was tested in the real world.

Each such robot was given a single real-world trial in the Winding maze. The progress of the robot in each real-world trial was measured by the proportion of the three turns it traversed successfully (i.e. a robot that completed two of the turns in a particular trial would receive a score of $\frac{2}{3}$). This measure was then averaged across all transferred robots for a given approach. The resulting quantity estimates the expected progress a transferred robot will make in the maze for each approach. Figure 10 illustrates this measure for all of the trained setups.

The best method at transferring, Reactivity + 30% Noise, progresses 63% of the way through the maze for an average transferred controller, significantly outperforming (Student's t-test; $p < 0.05$) all of the other approaches except Reactivity + 20% Noise (which progressed on average 50% of the way through the maze). Similarly, Reactivity + 20% Noise outperforms all of the traditional approaches (the Noise and Standard setups). Reactivity + 10% Noise and Reactivity alone are only significantly better than the 30% Noise setup (which progressed only 1.65% through the maze on average), and the traditional approaches are never significantly better than any of the Reactivity setups nor do

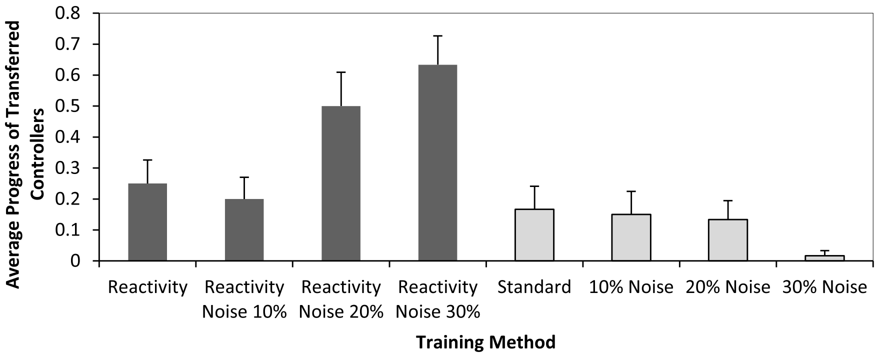


Figure 10: **Real World Transfer Results in Winding Maze.** The effects on transferability, i.e. the expected percentage of progress through the Winding maze for a robot transferred from simulation to the real world, are shown for training with various ER setups.

they significantly differ from each other. Interestingly, the transferability to the real world tends to decrease with increased noise levels in training when using standard Noise training (30% Noise never successfully traverses the entire maze in reality), but *increases* with increasing noise in training when paired with Reactivity (Reactivity + 30% Noise successfully traverses the maze in 45% of attempts).

Videos of transfers from simulation to the real world, including both reactive success and typical non-reactive failures, can be viewed at: `http://goo.gl/Qn9nz`. A typical robot trained with reactivity keeps an adequate distance to the surrounding walls. On the other hand, robots trained without reactivity often collide with the walls when trying to take a left or right turn.

5.4 Transfer to Novel Environments

To probe the limits of the generality of evolved behaviors, controllers can be transferred into mazes to which they had no exposure during training. That is, ideally controllers would be robust not only to varying levels of noise but also to structural variations in the domain. Of course, it may be unrealistic to expect that exposure to only one environment in training can induce perfect generalization across all possible environments. However, such a test might still provide interesting perspective on a method's ability to induce robustness.

Thus, to explore this idea, the same controllers that were transferred from simulation to reality in the Winding maze (as described in the previous section) were additionally transferred in simulation to other maze domains. That is, the most robust controllers trained in the Winding maze were evaluated on the simulated generalization test in the Straight maze, the Zigzag maze, and the

Deceptive maze. The Winding maze was chosen because it was also tested in real-world transfer, and many of its features are also present in the other mazes. That way, a controller evolved only to solve the Winding maze might be able to exploit the same underlying regularities in the other mazes to solve them as well.

Figure 11 shows how training through different approaches affects the probability of successful transfer to novel environments. In the Zigzag maze, candidates evolved with the Reactivity + 30% Noise setup are significantly more likely to transfer than any of the other setups ($p < 0.05$; Student's t-test), and in the Straight maze, Reactivity + 30% Noise is significantly better than the Standard setup or the 20% or 30% Noise alone setups. There were no significant pairwise differences in the Deceptive maze, because nearly all transfers failed completely; however, aggregating data reveals some benefit for reactivity. In particular, over all reactive transfers in the Deceptive maze, 11 passed the generalization test more than 10% of the time, while only 2 from the setups without reactivity did. This difference is significant ($p < 0.05$; Fisher's exact test), as is the pairwise comparison of average probability of success in the Deceptive maze if the data is aggregated by whether or not the setup includes reactivity ($p < 0.05$; Student's t-test). Furthermore, examining pairwise differences across all three transfer scenarios, setups including reactivity are never significantly worse at transferring than those without reactivity, and are often better (in 12 out of 48 such comparisons).

While the tests in entirely different mazes show that robustness gained through reactivity has natural limits (particularly in the Deceptive maze), the results overall show that in reasonable reproductions of the training environment (such as real-world reproductions), significant robustness does emerge. Thus given realistic expectations on possible deviations from training, reactivity can act as a useful tool for emphasizing robustness, such as in crossing the reality gap.

5.5 Fine-grained Generalization Test

To further investigate robustness, a fine-grained generalization test was devised to measure how well an evolved controller performs on individual levels of noise. In other words, this generalization test provides a more granular view of robustness, to demonstrate what levels of noise proved most challenging for each approach. The main idea is to measure how varying the level of noise affects a controller's performance.

For this generalization test, an individual was evaluated on 50 independent trials with a fixed level of sensor and effector noise (i.e. the level of sensor noise is the same as that of effector noise, and this level is constant for all 50 trials). For each run, the individual scoring the overall highest on this test from sampling the population at regular intervals is recorded, and averaged over each of the 20 runs. In all of the mazes except the Deceptive maze, the population is sampled every 100 generations. In the Deceptive maze experiment every 200 generations is sampled because of its longer duration. The generalization test

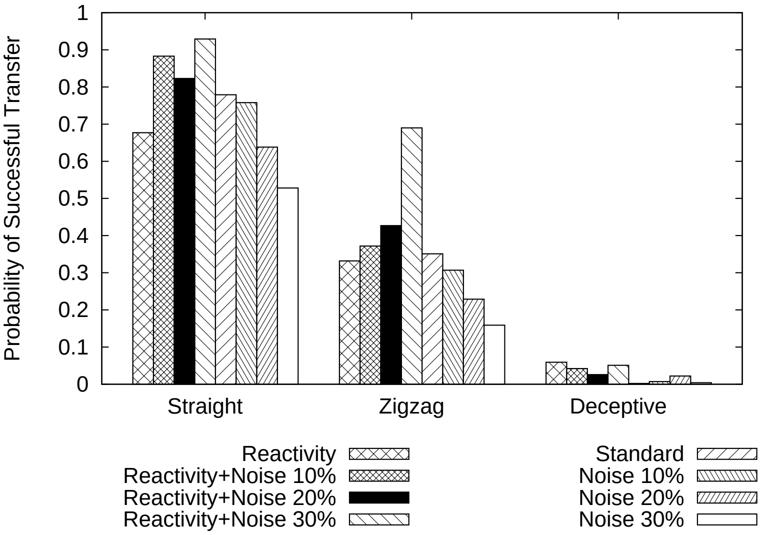


Figure 11: **Generalization to Novel Domains.** The effects on transferability to novel simulated domains from the Winding maze are shown for training with various ER setups. The main conclusion is that such transfer can sometimes be aided by reactivity training, although successful transfer may be more likely when tested domains sufficiently resemble those encountered in training.

itself is repeated with noise distributions from 0% to 35% at 5% intervals (for eight total testing noise levels). This approach to testing gives a sense of what levels of noise proved most challenging for each approach.

To assess statistical significance on the generalization test for each domain, a one-way ANOVA test was first applied across the five experimental setups for each level of generalization noise to demonstrate that the distributions are significantly different (at least $p < 0.05$). If at a particular noise level this first test was passed, then Student's t-tests were applied to measure the significance (assuming a p-value of 0.05) of pairwise differences between Reactivity and the other experimental setups.

Interestingly, figure 12 shows how the overall most-general (figure 8) combination, Reactivity + 20% noise, compares to the more traditional setups for robustness training in the Deceptive maze. The results in this maze most clearly demonstrate how traditional noise training can result in controllers over-fit to the training level of noise, e.g. the 30% Noise setup is most effective among the Noise setups only when tested at 30% or 35% noise. In contrast, the performance of Reactivity + 20% Noise degrades more uniformly when tested on increasing levels of noise.

Finally, the overall best performing method in the fine-grained generalization test, Reactivity + 20% noise, is never significantly worse and often significantly better than all of the Standard and the 10%, 20% and 30% Noise setups.

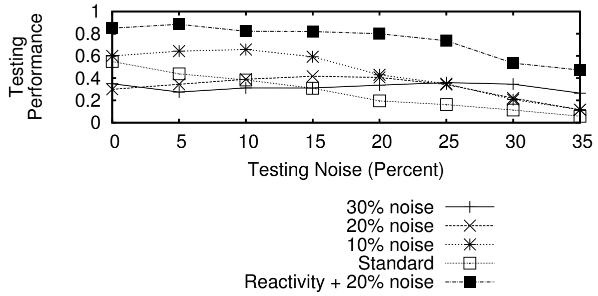


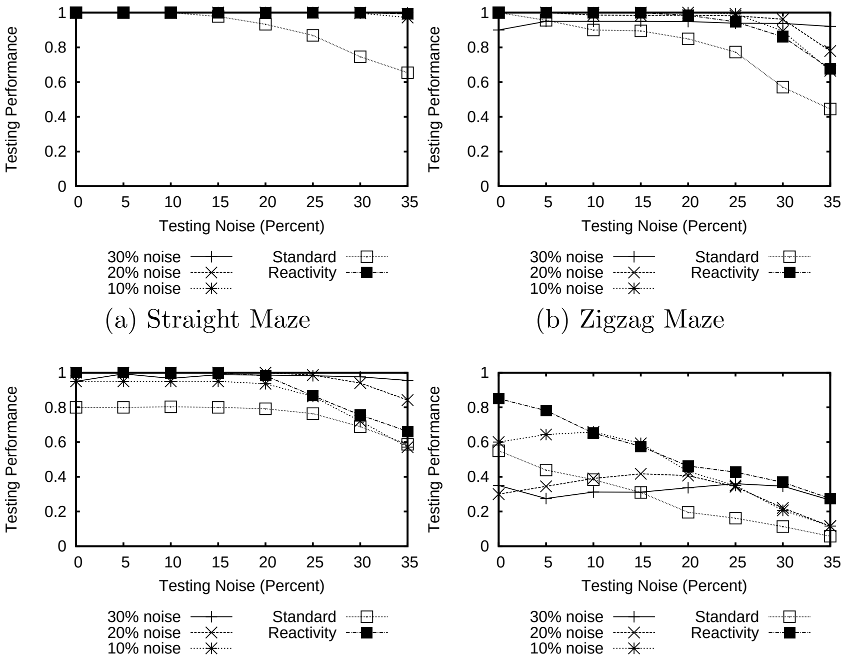
Figure 12: **Fine-grained Generalization Test Results in the Deceptive Maze.** The average probability of the best individual from a run to solve the generalization test at various levels of noise is shown for the Standard, Noise, and Reactivity + 20% noise setups for the Deceptive maze.

How the setup of Reactivity alone compares to the non-reactive setups (Standard and the three Noise setups) on the fine-grained generalization test is shown in figure 13. Over the five compared setups (Reactivity, Standard, and three different *training* noise levels) with eight *testing* noise levels each, there are 40 total generalization scenarios per domain.

Interestingly, in the Straight maze, the Standard setup (which has no exposure to noise nor any incentive to encourage reactivity) is the only setup that suffers as testing noise levels increase, supporting the motivation for this maze. More importantly, over all four domains, training with the Reactivity setup was never significantly worse at generalizing than training with the Standard setup, and was significantly better in 15 out of the 32 pairwise comparisons. Training with the Reactivity setup was significantly better at generalizing than the Noise setups in 7 out of 96 comparisons while Noise also was significantly better than Reactivity in 7 pairwise comparisons. Interestingly, the occasional significant advantages for the Noise setups only occurred when the noise level in the generalization test was 25% or greater, which suggests that reactivity training without noise may generally be most advantageous when it is likely that a robot will encounter only moderate levels of noise in reality.

Figure 14 compares the effect on fine-grained generalization of training with Reactivity + 10%, 20%, and 30% noise. The results demonstrate that *training* combining Reactivity with Noise demonstrates an advantage over training with Reactivity alone that is rarely dependent on a particular *testing* level of noise.

Finally, independently of the added noise level (e.g. 10%, 20%, or 30%), the Reactivity + Noise setups always perform the same or better than Reactivity without noise in all domains except in one scenario (Reactivity + 10% Noise in the Straight maze evaluated at 35% noise). The overall best performing method in the fine-grained generalization test, Reactivity + 20% noise, is never significantly worse and often significantly better than all of the Standard and the 10%, 20% and 30% Noise setups.



(a) Straight Maze

(b) Zigzag Maze

(c) Winding Maze

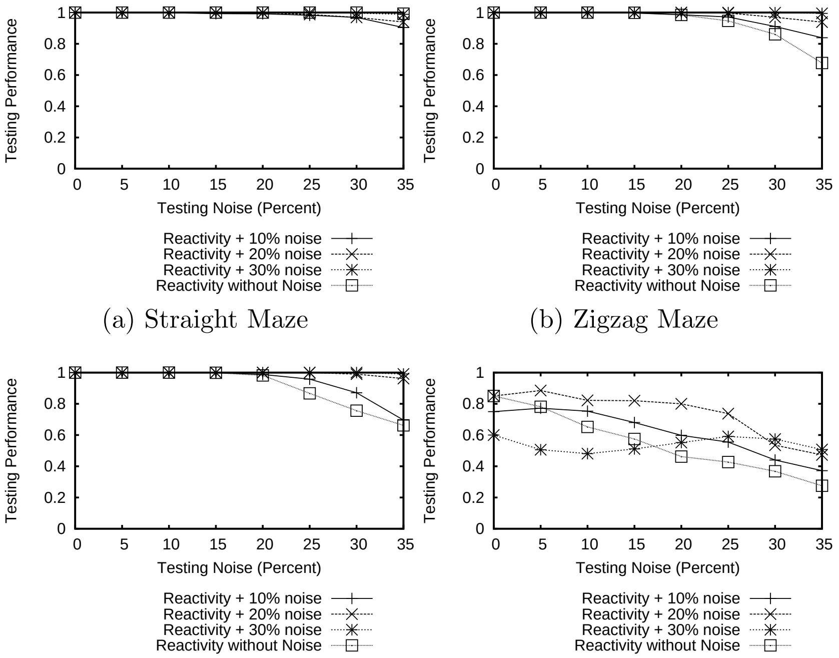
(d) Deceptive Maze

Figure 13: **Maze Navigation Fine-grained Generalization Test Results.** The average probability of the best individual from a run to solve the generalization test at various levels of noise is shown for different training methodologies over the four maze domains. The main result is that training with reactivity in all four domains is never significantly worse than training with noise (10%, 20%, or 30%) on the generalization test at moderate levels of noise (< 25%).

6 Discussion and Conclusions

This study demonstrates that the selection of ANN controllers based on the biologically-inspired concept of reactivity can produce robust controllers that do not depend explicitly on the specific training model both in simulation and when transferred to real robots. Solutions trained with reactivity as a goal had the best success rate when run in real robots and tended to perform best across *all* noise levels when tested in simulation. Solutions evolved with multiple noisy trials showed promise when tested on *specific* noise levels, implying that if a simulation is properly tuned to the actual conditions a robot will experience, then these approaches can be beneficial. However, such perfect tuning is difficult if not impossible to achieve in the real world, and these results show it may not be necessary.

Perhaps most interesting is that the ANN controllers that performed best



(a) Straight Maze

(b) Zigzag Maze

(c) Winding Maze

(d) Deceptive Maze

Figure 14: **Reactivity + Noise Maze Navigation Fine-grained Generalization Test Results.** The average probability of the best individual from a run to solve the generalization test at various levels of noise is shown for different variations of reactivity with noise over the four maze domains. The main result is that training with Reactivity + 20% noise is never significantly worse than training with any of the other setups (including Standard and the three Noise setups) and is often significantly better.

in the generalization test and transferred best to real robots were those that were trained with a combination of reactivity and a single noisy trial. Such a combination seems counter-intuitive because the robot is exposed only to a single, partly random experience of the world. However, that is also the case for organisms in the real world: all animals are born with noisy sensors and effectors and must learn the appropriate responses to them. A critical advantage that the controllers in these experiments have is that during simulation, the true signal is *known* to the experimenters, so the controllers can be directly rewarded for reacting to it instead of the raw noisy input. Thus robots must probe their environment not only for information about it, but also for information about their own sensors and effectors.

In ER, training with multiple noisy trials has the obvious effect of increasing computational costs linearly with the number of trials performed; thus reactivity provides a clear benefit by still producing robust controllers with only a single trial. However, a less obvious byproduct of traditional noisy trials is their effect on the evolutionary *search space*, that is, how they affect the search process for the solution. In training, single trials with reactivity were always no worse, and typically better than multi-trial approaches, confirming studies (Beyer, 2000) that suggest a deleterious effect of multiple trials of noise upon search, e.g. by obscuring less robust stepping stones that still lead to better behavior.

Rewarding reactivity is inspired by the tendency of animals to explore their environment when faced with unfamiliar circumstances. This natural response is intuitive, yet is ignored by some learning approaches in favor of finding an "optimal" training solution, even if that solution may not actually produce the best real-world results. In addition to providing a logical quantification for this concept of reactivity, this work suggests that there exist effective alternative approaches to multiple trials (Jakobi et al., 1995; Pinville et al., 2011) or real-world evaluation (Koos et al., 2011; Zagal and Ruiz-Del-Solar, 2007) for creating robust controllers that can cross the reality gap. More deeply, the results suggest that building and learning from an accurate model of the environment may not be the most important factor in attaining robust behavior. Rather than finely tuned, robust behaviors may simply be highly cautious and reactive, thereby precluding the need for perfection at every step.

7 Acknowledgements

This research was supported by DARPA and ARO through DARPA grant N11AP20003 (Computer Science Study Group Phase 3), and US Army Research Office grant Award No. W911NF-11-1-0489. This paper does not necessarily reflect the position or policy of the government, and no official endorsement should be inferred.

References

Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *The European Physical Journal B*, 63(3):329–339.

Balakirsky, S., Carpin, S., Dimitoglou, G., and Balaguer, B. (2009). From simulation to real robots with predictable results: methods and examples. *Performance Evaluation and Benchmarking of Intelligent Systems*, pages 113–137.

Baldwin, J. (1896). A new factor in evolution. *The American Naturalist*, 30(355):536–553.

Berlyne, D. (1966). Curiosity and exploration. *Science*, 153(3731):25–33.

Beyer, H. (2000). Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice. *Computer Methods in Applied Mechanics and Engineering*, 186(2):239–267.

Bongard, J. and Lipson, H. (2004). Once more unto the breach: Co-evolving a robot and its simulator. In *Proceedings of the International Conference on Artifical Life (ALIFE9)*, pages 57–62.

Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121.

Bongard, J. C. and Pfeifer, R. (2003). *Evolving complete agents using artificial ontogeny*, pages 237–258. Morpho-functional Machines: The New Species (Designing Embodied Intelligence). Springer-Verlag.

Brooks, R. (1994). Artifical life and real robots. In *Proceedings of the European Conference on Artificial Life*, pages 3–10.

Clune, J., Stanley, K., Pennock, R., and Ofria, C. (2011). On the performance of indirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation*, 15(3):346 –367.

Coello, C. A. (1999). A comprehensive survey of evolutionary-based multi-objective optimization techniques. *Knowledge and Information Systems*, 1(3):129–156.

Coello, C. A. (2006). Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine*, 1(1):28–36.

Cyberbotics (2012). Webots. Commercial Mobile Robot Simulation Software.

D'Ambrosio, D., Goodell, S., Lehman, J., Risi, S., and Stanley, K. O. (2012). Multirobot behavior synchronization through direct neural network communication. In *Proceedings of the International Conference on Intelligent Robotics and Applications (ICIRA 2012).*

D'Ambrosio, D., Lehman, J., Risi, S., and Stanley, K. (2011). Task switching in multiagent learning through indirect encoding. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 11).* IEEE.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Drchal, J., Kapral, O., Koutník, J., and Šnorek, M. (2009). Combining multiple inputs in hyperneat mobile agent controller. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN 2009)*, pages 775–783. Springer-Verlag.

Gauci, J. and Stanley, K. O. (2010). Autonomous Evolution of Topographic Regularities in Artificial Neural Networks. *Neural Computation Journal*, 22(7):1860–1898.

Glickman, S. and Sroges, R. (1966). Curiosity in zoo animals. *Behaviour*, pages 151–188.

Gomez, F. and Miikkulainen, R. (2004). Transfer of neuroevolved controllers in unstable domains. In *Proceedings of the Genetic and Evolutionary Computation Conference*. Springer.

Green, C. (2003–2006). SharpNEAT homepage. `http://sharpneat.sourceforge.net/`.

Haasdijk, E., Rusu, A., and Eiben, A. (2010). Hyperneat for locomotion control in modular robots. *Evolvable Systems: From Biology to Hardware*, pages 169–180.

Hagen, E. and Hammerstein, P. (2005). Evolutionary biology and the strategic view of ontogeny: Genetic strategies provide robustness and flexibility in the life course. *Research in Human Development*, 2(1-2):83–97.

Harvey, I. (1993). *The Artificial Evolution of Adaptive Behavior*. PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, Sussex.

Hornby, G. S. and Pollack, J. B. (2002). Creating high-level components with a generative representation for body-brain evolution. *Artificial Life*, 8(3):223–246.

Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160:106–154.

Iba, H. and Terao, M. (2000). Controlling effective introns for multi-agent learning by genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 419–426.

Jakobi, N. (1998). *Minimal Simulations For Evolutionary Robotics*. PhD thesis, University of Sussex.

Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. *Advances in artificial life*, pages 704–720.

Jensen, M. (2003). Guiding single-objective optimization using multi-objective methods. *Applications of Evolutionary Computing*, pages 91–98.

K-Team (2010). Khepera III mobile robot. *http://www.k-team.com*.

Knoester, D., Goldsby, H., and McKinley, P. (2010). Neuroevolution of mobile ad hoc networks. In *Proceedings of the Conference on Genetic and Evolutionary Computation*, pages 603–610. ACM.

Knowles, J., Watson, R., and Corne, D. (2001). Reducing local optima in single-objective problems by multi-objectivization. In *Evolutionary Multi-Criterion Optimization*, pages 269–283. Springer.

Koos, S., Mouret, J., and Doncieux, S. (2010). Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 119–126. ACM.

Koos, S., Mouret, J.-B., and Doncieux, S. (2011). The transferability approach: Crossing the reality gap in evolutionary robotics. *Evolutionary Computation, IEEE Transactions on*, 17(1):122–145.

Langdon, W., Soule, T., Poli, R., and Foster, J. (1999). The evolution of size and shape. *Advances in genetic programming*, 3:163.

Lehman, J., Risi, S., D'Ambrosio, D. B., and Stanley, K. O. (2012). Rewarding reactivity to evolve robust controllers without multiple trials or noise. In *Proceedings of Artificial Life Thirteen (ALIFE XIII)*.

Lehman, J. and Stanley, K. O. (2011). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223.

Lipson, H. and Pollack, J. (2000). Automatic design and manufacture of robotic lifeforms. *Nature*, 406(6799):974.

Martin, A. P. (1999). Increasing genomic complexity by gene duplication and the origin of vertebrates. *The American Naturalist*, 154(2):111–128.

Matarić, M. (1997). Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83.

Matarić, M. and Cliff, D. (1996). Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, 19(1):67 – 83.

Michel, O. (2004). Webots: Professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):39–42.

Miglino, O., Lund, H., and Nolfi, S. (1995). Evolving mobile robots in simulated and real environments. *Artificial life*, 2(4):417–434.

Mouret, J.-B. and Doncieux, S. (2012). Encouraging behavioral diversity in evolutionary robotics: an empirical study. *Evolutionary Computation*, 20(1):91–133.

Ng, A., Coates, A., Diel, M., Ganapathi, V., Schulte, J., Tse, B., Berger, E., and Liang, E. (2006). Autonomous inverted helicopter flight via reinforcement learning. *Experimental Robotics IX*, pages 363–372.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics*. MIT Press, Cambridge.

Ollion, C., Pinville, T., and Stephane, D. (2012). With a little help from selection pressures: evolution of memory in robot controllers. In *Proceedings of Artificial Life Thirteen (ALIFE XIII)*, volume 13, pages 407–414.

Pinville, T., Koos, S., Mouret, J.-B., and Doncieux, S. (2011). How to promote generalisation in evolutionary robotics: the ProGAb approach. In *Proceedings of the Conference on Genetic and Evolutionary Computation*, pages 259–266.

Revello, T. and McCartney, R. (2000). A cost term in an evolutionary robotics fitness function. In *Proceedings of the Congress on Evolutionary Computation*, pages 125–132. IEEE.

Shannon, C. (1949). A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423.

Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.

Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based indirect encoding for evolving large-scale neural networks. *Artificial Life*, 15(2).

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:99–127.

Stanley, K. O. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.

Stanley, K. O. and Miikkulainen, R. (2004). Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21(1):63–100.

Udin, S. and Fawcett, J. (1988). Formation of topographic maps. *Annual Review of Neuroscience*, 11(1):289–327.

Verbancsics, P. and Stanley, K. O. (2010). Evolving static representations for task transfer. *Journal of Machine Learning Research*, pages 1737–1763.

Woolley, B. G. and Stanley, K. O. (2010). Evolving a single scalable controller for an octopus arm with a variable number of segments. In Schaefer, R., Cotta, C., Kołodziej, J., and Rudolph, G., editors, *Proceedings of Parallel Problem Solving from Nature*, pages 270–279. Springer.

Zagal, J. C. and Ruiz-Del-Solar, J. (2007). Combining simulation and reality in evolutionary robotics. *Journal of Intelligent & Robotic Systems*, 50(1):19–39.

Zagal, J. C., Ruiz-del Solar, J., and Vallejos, P. (2004). Back to reality: Crossing the reality gap in evolutionary robotics. In *IFAC Symposium on Intelligent Autonomous Vehicles*.

Zufferey, J., Guanella, A., Beyeler, A., and Floreano, D. (2006). Flying over the reality gap: From simulated to real indoor airships. *Autonomous Robots*, 21(3):243–254.